Innovation Capabilities in Software Industry through the Interfaces Creation among Users and Producers: The Case of Mexican Firms

Jose Luis Sampedro Hernandez

Key words: Interface, Knowledge, Innovation Capabilities, Software Firms.

I. Introduction

The aim of this paper is to describe and analyze the interface construction process among software firms and other social actors and how, through the interface, the software firms create technological knowledge and innovation capabilities. The central research question I will try to answer is about: How do Mexican software firms build up interfaces with other social actors and why it is important in the knowledge generation process and in the creation of innovation capabilities?

The cases to be studied of this research will be Mexican firms which develop and design 'customized-software'. Some of them have created technological knowledge and accumulate technological capabilities through simple and stable interfaces, in which the information flow predominates on the knowledge flow. Others of them have created complex interfaces, in which predominate the knowledge flow and a high level of interaction among user and producer firms. In these terms, the *user-producer interaction* supports the firms to create new knowledge and increase the innovation capabilities. The Mexican firms have developed and designed proprietary software (PSW), however, there are firms that have migrated to the development and design of open-source software (OSW). The firms that develop and design OSW have at least three opportunities in a short term: a) to create new technological knowledge, b) to accelerate the innovation capabilities accumulation, and c) to open and to diversify the software market in a 'transversal' way.

This paper is organized in 5 sections. After this introduction, section 2 presents the principal bodies of literature. Next section explains the research strategy has been followed in this paper. Section 4 describes the Mexican Information Technology Industry. Finally, section 5 presents important results and conclusions.

II. Analytical Framework

To tackle the research proposal I have incorporated three bodies of literature. The first one is about the creation of interfaces, the second one is about the generation of knowledge, and the third one is about the technological capabilities by firms in developing countries.

II.1 Interface as starting point in the knowledge generation process and innovation capabilities

Andersen (1991, and 1996), Lundvall (1985, 1988 and 1992), Andersen and Lundvall (1988), and Fagerberg (1993) have studied the form in which firms create technological knowledge through the interaction among different actors such as users, buyers, universities, customers, and so on.

The concept of interface is not very different of user-producer interaction concept defined by Lundvall (1985, 1988, and 1992). The concept of user-producer interaction emphasises on the relevance of interactive learning among the agents. The producers learn doing products (learning by doing) and the users learn using it (learning by using), the interaction among users and producer lets them exchange their experiences (learning by interactive). The agents exchange technological information, knowledge and know-how about products, process and organizational models. The interaction implies that the learning by interactive lets to users and producers increase their innovation capacity and competitiveness.

In base on Lundvall (1988), the user-producer interaction is based on different types of relations that depend on: i) symmetric and asymmetric relations among the agents, ii) standardization of the technology, frequency and duration of exchange, iii) economic and cultural context, iv) organizational dimension, and v) knowledge flows inside and outside the firm. This scholar studies the interactive process at different levels, one of them is at microeconomic level and the other one is at national innovations system level.[1] Andersen (1991) conceptualizes the user-producer interaction under a techno-economic paradigm context at micro level,[2] and he calls interface to the user-producer interaction.

Interface and TEP with a microeconomic approach

In base on Andersen (1991:121), in the technological change context, the techno-economic paradigm (TEP) means coordination among producers and users of different kinds of artifacts functionally defined which are exchanged in the market. He defines interface as "a relationship between two agents in which there are different kinds of information flows". If an interface is simplified and standardized the information necessary for each one of the agents will be delimited. However, the innovation process presupposes an information-rich interaction and thereby non-standardized interfaces but complex interfaces.

The concept of interface has idea about stability among users and producers. This point has been developed under two principles of interfaces designing. On the one hand, the principle of *commodity abstraction* minimizes the necessity of information flows, where simple interrelations and relatively stables are of fundamental importance on function of economic system. Under this principle is possible to create routines among the agents that are involved in the interface; if the routines are developed under the assumption of stable interfaces will have a few possibilities to change the strategy. One technology can be maturity and standardized, and the interface can be created and fixed for a long time. In this sense, it is possible to create a path of maintenance and evolution.

---

[1] Lundvall (1985, 1988, and 1992) explains the conditions microeconomic and institutional which are necessaries in the National Innovation System.

[2] Perez (1986, 2002 y 2003) and Freeman y Perez (1988) define TEP at macro level, while Andersen (1991, 1996) defines it at micro level.

On the other hand, the principle of interactive learning among producers and users is a kind of necessary interaction to create new products, but at the same time this principle increases the necessity of information flows. Under this principle the interface can not be stable because the interactive learning implies strong links and effects that are not part of economic exchange of commodities, unless common information is being included. For this reason the interfaces tend to be different.[3]

II.2 Interface and Knowledge

The TEP can be considered like a way to delimit the necessities of information flows among the agents, in that way it provides a greater condition to the existence of complex techno-economic systems. According to the Andersen's (1991, 1996) perspective, the interplay among users and producers has a fundamental role in the definition of technological paradigm. The relevant problems are the knowledge base and the process about productive efficiency. The firm's information is just one part limited of what exits in the environment (located knowledge), and is part of a system of techno-economic knowledge based on an extended division of labor between different knowledge-holding units (Andersen 1991:128).

Tacit knowledge versus codified knowledge

In base on Nonaka and Takeuchi (1994:65) the firm's knowledge contains tacit and codified elements that are distributed in different areas. The tacit knowledge is personal and from specific context, it is difficult to formalize and translate. Explicit knowledge can be translated through formal and systematic means such as publications, patents, and so on.

The technology's characteristics affect the form in which the firms can access to relevant knowledge. If the knowledge is tacit, complex, and is part of a complex system, the informal means such as work circles and training are more important to transmit the knowledge. In these conditions, the geographic proximity among the agents is important to guarantee a suitable knowledge transference and accumulation. But, if the knowledge is standardized, codified, simple and independent, the formal means such as internal publications (to spread the firm's works), product and process manuals and patents are more important to transmit the knowledge. In these conditions, the geographic proximity is not fundamental to transmit the knowledge (Maleaba y Orsenigo, 1990, 1995 y 1996). It means that the knowledge could be codified in one context but tacit in other one; as well as complementary (Senker y Faulkner, 1996).

II.3 Technological Capabilities Accumulation

Bell (1984), Pavitt (1984), Bell and Pavitt (1993, 1995), Lall (1987, 1992, 2000), Dogson (1993), Katz (1984), Hobday (1995, 2000, 2001) Dutrénit (2000), and Vera-Cruz (2004) have proposed an analytical framework which draws on a taxonomy of technological capabilities for the

[3] Not many interfaces are important in the development of new products. Some users play an active role and create new products. Others obtain the benefits from the incremental innovation of the products, but they do not play an active role in the innovation process. And others users accept the standardized products that are in the market, even though the products are not appropriate for their necessities, and then they accept the principle of commodity abstraction (Andersen, 1996:55).

manufacturing industry in developing countries, originally proposed by Bell and Pavitt (1995). The basic idea is that capabilities represent abilities to do things, and technological capabilities reflect the dominion of technological activities. Based on empirical research at firm level, this literature has elaborated taxonomies to describe the gradual processes of accumulation, from a stage that reflects minimum levels of knowledge (needed for the routine operation) to the stage of advanced innovative capabilities.

In technical, economic and social activities there are different interfaces created among different agents. From the technical change approach only the relevant interfaces contributes to improve and increase the innovation capacity to create new product, service and process. In that sense, the concept of interface is of fundamental importance to understand the nature of interaction and links that one firm could create with different agents. The software firms do not act in an isolated way and they create simple or complex interaction with different agents. In these kind of interactions is possible to generate information flows and in some cases knowledge flows as well. When the firm can assimilate great knowledge flows and there are suitable mechanisms (like an interface) to transform that flows in new knowledge, then the firm can accumulate innovation capabilities, the reason is because the creation of interfaces could allow to the firms accumulate better capacities.

III. Research Methodology

Both the nature of the central research question described in the introduction and the bodies of literature suggests as better research strategy to the multiple-case study. Through qualitative methods one can understand the reality like others experience it (Yin, 1994:5; Phillips y Pugh, 2001:70).

The research strategy is based on exploratory multiple-case study. The *cases* to be studied are software firms. The *unit of analysis* refers to the interface among software firms and other social actors. The type of information analyzed is qualitative and the results are not sustained on the case (firms) but the unit of analysis. The empirical analysis is sustained over 17 interviews at project leaders and software developers in 14 software firms producing both proprietary and open-source software. The field-work was made between March 2004 and March 2005. The firms interviewed are located in Mexico, Guadalajara, Leon, and Monterrey City.

IV. Mexican Software Industry

During the period of 1992 to 2003 the Manufactured Software Industry participated with 7.2% of production in the total Information Technology Industry (IT) in Mexico. Within the IT industry one of the most important industries is the hardware industry which has participated (in average each year) with 37% of total production. The whole IT industry contributes with less than 1% of the GNP,[4] while the software industry contributes with 0.1% in the GNP (AMITI, 2003). The demand of products and services is concentrated in retail services, finance, electronic and automotive industries, leaving out the small and medium-size businesses.

[4] Gross National Product.

There are specific programs such as PROSOFT, which is trying that the software industry develops their industrial and technological capabilities. PROSOFT attempts to increase the technological and industrial capabilities, both inside and outside the country; in this way Mexico becomes the Latin-American leader in software. At the end, the economic goal is to export 5 billion of dollars each year starting 2013 (Secretaría de Economía, 2001-2006). The **financial** resources to develop the software industry are 15 millions of dollars each year.

IV.1 Proprietary software segment

Nowadays, there is not a characterization of the Mexican software industry, the data in terms of number of firms, number of employees, and sales are not exact. In base on interviews, in Mexico there are between 1000 and 1500 firms formally established, nevertheless, some studies show us some idea about the structure that could present the software industry. A survey made by AMITI (2001) to 206 firms of proprietary software shows an atomistic structure: 87% of firms are micro and small with less than 30 employees in average, 6.7% are medium firms, 5.3% are big firms and just there are two corporative with approximately 1500 employees each one.

IV.2 Open-source software segment

In Mexico there is not a formal open-source software industry. In base on interviews, there are 100 firms (more o less) that development and design open-source software. Almost 90% of the firms are micro with less than 15 employees each one. The development of this kind of software is made in ‘developer-communities’, in which the principal agents are developers and ‘aware users’. In 2004 were 1000 developers and almost 10 thousand aware users at national level, working in different application and tools, and in different communities.

In spite of their poor development, the production of applications made in open-source software has increased in the last years, mainly for the servers segment. In 2003 were 14,960 servers with open-source software, and in 2004 it increase to 20,000 units at national level. The market still is limited, but, in 2002 the open-source software industry represents 7.9 percent and in 2004 it represents 9.2 percent at national level.

V. Important Results

In this work, interface is defined as a social device to translate information flows in new knowledge among two or more actors. The innovation process in the Mexican software industry is encouraged starting different types of interfaces developed through different social devices such as the user-producer interaction and links between software firms and universities. These are two of the better social devices to translate information flows in new knowledge. There is other social device in the context of open-source software called ‘developer-communities’ of open-source software. The principal elements in each interface are the information and knowledge.

The interfaces have elements of context such as the great dynamism in the software industry which changes quickly, I mean, the using of new applications in different industries implies a change in the techno-economic paradigm at micro level.

It is possible to identify an interface through its simplicity or complexity in terms of different kind of interaction among the firms and other agents. The Mexican software firms create three types of interfaces: simple, standardized, and complex. The principal agents that are involved in the interface are: passive and active users, aware users, universities, and training centers.

V. 1 Simple, standardized and complex interfaces

Simple interface: simple relation between user and producer

The simple interface is shaped and determined by the information that exists in the codified environment. In these terms the producer can find in an easy way the information about productive and organizational process to develop software (applications) that the users need. When the user is part of a maturity industry there are great mounts of information and codified knowledge about the organizational and productive processes, and the software firm can access to that information through of magazines, seminars, specialized books, and others.

Through simple interface the software firm can translate information and codified knowledge in new products and services, the software applications are useful to administrate and organize the processes.

The users are small and medium-size firms from manufacture and service sectors, however, the software applications that they use are not very different. Some times the software applications needs to be adapted to necessities of each user, but, is just a minor modification. This means that the relation between producer and user is passive because of an user needs relatively simples and great mounts of information about requirements and necessities which it is possible to find in the codified environment (for instance Internet). The relation is limited to purchasing-selling contract. In other cases, the relation between software firm and user implies adaptation or upgrading of one application, or consulting services as well.

The activity of the simple interface consists on trade activity, marketing, maintenance of systems that have been installed in the firm, personal sales, or training to use the software application. Few times, there are projects that involve with design and full development of software programs.

If the project implies activity trade or personal sales the interaction level between producer and user is low, the sessions are between 6 to 10 hours by 3 or 5 days. If the project implies maintenance of systems or training to use the software application interaction level is between 15 days and 2 months. I mean, the relation is informal; it is just formal in the case of purchasing-selling contract.

Standardized interface:

A standardized interface is designed and determined depending on kind of project. The firms are located in organized markets, in such a case, to develop of any application can be generalized to more than one firm. The codified environment can provide some necessities of information to develop software that the user needs, although, the producer will require some specific information from the user really needs. So, in this interface could have a deep user-producer interaction because the producer would not find the specific information in the codified environment and will need for the user to get specific information.

Through standardized interfaces is possible to translate information and knowledge from users to create new products, services, and processes very specific. The user is active and not passive like in the simple interface. The basic information about user's necessities could be found in the codified environments but not core information.

To show one example of standardized interface I will present the case of Degas. Degas is a medium-size firm that develops software to leather-tanning sector. The Degas' owners had worked in a leather-tanning firm before they created Degas, they know the productive ad organizational process from the users. Degas finds information and codified knowledge in the codified environment, but specific information and knowledge is incorporated in the people that worked for user firm. For this reason, the user-producer interaction is important because of, there is a tacit knowledge that the producer does not know it. In this process, there is a high level of interaction. A good design of the project is calibrated by the user, and the software program will be efficient if the user-producer interaction is deep.

The user will need to get a proper training to use a software program, I mean the producer trains to users. Few users have used software program in their productive areas, but through the system designed by Degas the user could improve their productivity, reduce costs, and increase the production of leather. One project last 1000 hours or between 8 and 16 months, it depends on the size firm. At the beginning of the project the interaction is formal but at the end it is informal.

Complex interface:

A complex interface is designed and determined by intensity of the user-producer interaction and the links between firms and universities. One of the most important differences among the three interfaces is the role that the user and universities play. In the complex interface the user and the university take an active and aware role in the interface creation process. The complex interfaces have a high content of knowledge and it is associated to the creation of new products and services. Also the developer-communities are an important social device to translate information to new knowledge, but the communities operate in the open-source software context.

In the complex interface the software firms need specific information to made applications. That information is difficult to find in the codified environment and the producer needs to know the necessities and requirements of user. The projects are very specific and they require design and development of new programs.

In many cases the users are pro-active and processes are interactive to develop applications. In other cases, such as in the open-source software context the user is 'aware' because has knowledge about software technologies, sometimes the user takes an active role and is more active that the producer. This type of user has know-how about the tools that the producer uses, and the user takes conscience about the maturity of the application that is being developed. At the same time, the active role of the aware user depends on maturity of the developer-communities.

Table 1
Characteristics about simple, standardized, and complex interfaces

Category	Simple interface	Standardized interface	Complex interface

[1] Agents *Extent of interaction between producer and agents*	Passive users *Low*	Active users *Medium*	Pro-active and aware users, universities *High*
[2] Complexity of the interaction* *Access to information about the requirements and necessities of the users* *Information and knowledge about the requirements and necessities of the users*	Sale-purchase relation, adaptation of software *Through Internet, Extranet, e-mail* *Simple information and codified knowledge*	Developments, testing and set up of software programs *Through Internet, Extranet, e-mail, face to face interaction* *General and specific information, tacit and codified knowledge*	Design and development of new software programs, Developer-communities, linkages firm-university *Specific information and specific and tacit knowledge*
[3] Stability of the software technology used in one project ***	High	Medium	Low and medium
[4] Interaction intensity ** *Mounts of information exchange* ***	Days *Low*	Months *Medium*	Months *High*
[5] Formal versus informal of the interaction****	Formal (+), based on sale-purchase relation	Formal (+), based on sale-purchase contract and training	Formal (+), based on training courses; informal (+)

* (i) Sale-purchase relation, (ii) adaptation of software, (iii) deign and development of new software programs, (iv) installation of software application, (v) maintenance, (vi) system integration.
** Days, weeks, months.
*** High, medium, low.
**** To record work sessions.
Source: Own elaboration based on interviews.

There is other kind of social device as the linkages among software firms and universities. The aim of the linkages between Goya (software firm) and the UNIVER (university) is to train the people in the university to obtain core-knowledge according to the necessities of the sector. The linkages between Goya and UNIVER last 2 years and the output was not just the creation of human capital but institutional capital as well. For instance, new firms have been created starting the linkages, I mean the result is an effect of spill over.

As the table 1 shows us, a complex interfaces could be created between software firms and pro-active or aware users, but not with passive users.

V.2 Knowledge creation

The knowledge created through different interfaces has levels of complexity and dynamism. In both simple and standardized interfaces, the information flows predominate on the knowledge flow. The type of knowledge is codified and it is spread in a formal means. The knowledge is created, systematized, and managed inside of firms. In spite of the problems in terms of rationalization of software, the firms of proprietary software tend to systematize its process, to measure the quality and the efficiency. In this way there are able to manage huge amount of information and knowledge. In a proprietary context the knowledge is created inside the firm through the user-producer interaction (as long as the user plays an active role); whereas, in an open-source context it is created inside the developer-communities. In the complex interface the knowledge predominates on the information flows and this information is translated in

knowledge through the user-producer interaction and by the linkages between the software firm and the university.

The general knowledge is created with different specialties such as information technology, computing systems, computing sciences, management systems, information systems, applied mathematics and computation, and so on. But the sector needs core-knowledge in the following specific-technologies: system infrastructures (middleware –legacy systems), multimedia 2D and 3D, Internet platforms, Linux-base software, software testing, inside and outside networks, and so on. The linkages among software firms and universities will improve the specific programs to create specific knowledge.

V.3 Innovation capabilities

The technological capabilities accumulation process is slow, but the Mexican software firms have the capacity to explore new market niches: high technology markets and traditional markets. It is important because of the software in traditional industries will improve their production capacity and the level of competitiveness national and international. The Mexican software firms have learned to develop software for different platforms such as mainframes, microcomputer, and so on, but it is important to mention as well, they learned to develop software to produce goods in other industries such as textile or leather-tanning. At the same time, there is evidence about how Mexican software firms have learned to manage and systematized the information and knowledge created inside the firm.

In the open-source and proprietary contexts the software process design does not have big differences, but it depending on the 'kind of software' is being developed. The proprietary context, for instance, it will be able to follow *commercial trend of action*, whereas, the open-source context can follow *moral or ideological* tendency *of action*, in this way, the design process can be different, or no, it depends on the type of project. Nevertheless, the development process is extremely different in both contexts. In a proprietary context the software is usually developed within the conceptual frame of the commercial secret (copyright), where the software that is developed with a powerful version in a short-run can be able to have a better opportunity to dominate a big part of the market. In this sense the protection of source-code is really importance. Also, there is proprietary software not-commercial, I mean, the finance element is not the only one factor that is taken into account to decide whether software is proprietary or free, it depends on producer decides to make public (or not) the source-code, and to guarantee its access and modification to all the people.

The open-source software is usually developed by two groups of people. The first one is a small group of leaders (project developers), who contribute to develop new codes or improve it. The second one is a big group of 'aware' users that contribute with code, ideas, and so on. Users and producers use to communicate by e-mail, and depending on complexity of project they can use sophisticate tools of collaborative development. The development process supports them to accumulate technological capabilities, more that the design process.
In a proprietary context the development process is more dynamic that in the open-source context. The reason is because firms are able to standardize the processes and to mange the knowledge created inside of the firm. In an open-source context, the development process follows

with the same *hand-made inertia* from decades ago, but with a better knowledge creation, which support them to increase its innovation capabilities.

Conclusions

1. The software industry is loaded of a strong dynamism where the things change quickly and the user-producer relationship needs to be dynamic. In this sense, the interfaces need to be modified constantly before solving specific problems of users. On the one hand, in Mexico there are many software firms on interfaces simples and standardized that could exist for a long time, but this implies that producers and users are passive in the dynamic market. In this situation the producer, for instance, could improve the sales, installing package software in the user firm, or give to the software systems just maintenance, but this pattern could reduce the innovation capacity.

2. On the other hand, there are few firms on complex interfaces. The difference with the other interfaces is that complex interfaces are for a short time. The user-producer interactive is active and dynamic but the things changes very quick and software Mexican firms do not be able to modify their behavior as the software industry require. The Mexican software firms need to improve their capacity to adapt to use new applications and to create new products for other markets such as manufacture sectors. In this sense, the creation of complex interface seems to have a better possibility to create new knowledge and not just information, but it depends on the maturity of technology to which the interface is associated.

3. However, there are software firms that have created capabilities to design new products, processes, or services for manufacture sectors. And this is a window of better opportunity because there are evidence that software firms have skills, know-how, and technological capabilities to do it.

4. Finally, the market niches that represents a better opportunity are located in sectors that demand manufacture software and not only administrative software (in which most of Mexican firms have been focused), and open-source software offers the possibility to access to new market niches, in a low cost of software development.

References

AMITI (2001). *Esquema de Apoyo Gubernamental a la Industria del Software*. AMITI, México.

Andersen, E. S. (1991). "Techno-economic Paradigms as Typical Interfaces between Producers and Users", in *Evolutionary Economics*, Vol. 1, Num. 2. USA:119-144.

Andersen, E. S. (1996). *Evolutionary Economics. Post-Shumpetian Constributions*. Biddles Ltd. Guildtord and King's Lynn, London.

Arthur, W. B. (1989). "Competing Technologies, Increasing Returns, and Lock-In by Historical Events", *The Economic Journal*, Vol. 99, Num. 394, March, pp 116-131.

Bell, M. (1984). "Learning and the Accumulation of Industrial Technological Capacity in Developing Countries", in Fransman and K. King (eds), *Technological Capability in the Third World.* Macmillan, London: 187-209.

Bell, M. y K. Pavitt (1995). "The Development of Technological Capabilities", in I. Haque (ed.) *Trade, Technology and International Competitiveness.* The World Bank, Washington: 69-101.

Dodgson, M. (1993). "Organizational Learning: A Review of Some Literatures". In *Organizational Studies*. Vol. 14, Num. 3. UK: 375-394.

Dosi, G. (1982). "Technological Paradigms and Technological Trajectories", *Research Policy*, Num 11.

Dutrénit, G. (2000). *Learning and Knowledge Management in the Firm: From Knowledge Accumulation to Strategic Capabilities*, Cheltenham: Edward Elgar, U. K.

Dutrénit, G. y A. O. Vera-Cruz (2001). "Aprendizaje, Conocimiento y Capacidades Tecnológicas", Monografía No 2 del proyecto *Aprendizaje Tecnológico y Escalamiento Industrial: Generación de Capacidades de Innovación en la Industria Maquiladora de México,* COLEF/FLACSO/UAM.

Eischen, K. (2002). *The Social Impact of Informational Production: Software Development as an Informational Practice.* Working Paper Series 2002-1 University of California, USA.

Freeman, C. and C. Perez (1988). "Structural Crises of Adjustment: Business Cycles and Investment Behavior", in Dosi *et al*., (eds.), *Technical Change and Economic Theory.* Pinter Publishers, London, UK.

Hobday, M. (1995). *Innovation in East Asia.* Edward Elgar, UK.

..... (2000). "East Versus Southeast Asian Innovation Systems: Comparing OEM and TNC led Growth in Electronics", in Kim and Nelson (eds.), *Technology, Learning, & Innovation Experiences of Newly Industrializing Economies,* Cambridge, UK.

..... (2001). "OEM vs TNC-led, Growth in Electronics: Comparing East and South East Asian Innovation System", en Dutrénit, Garrido y Valenti (eds.), *Sistema Nacional de Innovación: Temas Para el Debate en México*, UAM, México.

Lall, S., (1987). "The Acquisition of Technological Capability by India". In *Learning to Industrialize*. Macmillan, London**.**

.....(2000). "Technological Change and Industrialization in the Asian Newly Industrializing Economies: Achievements and Challenges", in Kim and Nelson (eds.), *Technology, Learning, & Innovation Experiences of Newly Industrializing Economies*, Cambridge, UK.

Lundvall, B. A. (1985). *Product Innovation and User-Producer Interaction.* Aalborg University, Press Aalborg.

…….. (1988). "Innovation as an Interactive Process: from User-Producer Interaction to the National System of Innovation", in Dosi, *et al*., (eds.), *Technical Change and Economic Theory*, Pinter Publishers, UK.

…….. (1992). *National System of Innovation. Towards a Theory of Innovation and Interactive*
Learning. Pinter Publisher, London.

Malerba, F. y L. Orsenigo (1990). "Technological Regimes and Patterns Innovation: a Theoretical and Empirical Investigation of the Italian Case", in Heertje A. (ed.), *Envolving Technology and Market Structure,* University of Michigan Press, USA.

…….. (1995). "Shumpeterian Patterns of Innovation", in *Cambridge Journal of Economics,* Num. 19, USA: 47-65.

…….. (1996). 'The Dynamics and Evolution of Industries', in *Industrial and Corporate Change*, Vol.5, Num. 1, Oxford University Press, USA.

Maxwell, P. (1981). *Technological Policy and Firm Learning in Less Development Countries: a Case Study of the Experience of the Argentina Steel Firm Acindar SA*. D. Phill, SPRU, Sussex. Cap. 2.

Nonaka, I. y H. Takeuchi (1994). *La Organización creadora de Conocimiento. Cómo las Compañías Japonesas crean la Dinámica de la Innovación.* Oxford, Londres.

Pérez, C. (1986). "Las Nuevas Tecnologías: Una Visión de Conjunto", en C. Ominami (ed) *La Tercera Revolución Industrial. Impactos Internacionales del Actual Viraje Tecnológico,* GEL, Argentina.

…….. (2002). *Technological Revolutions and Financial Capital.* Edward Elgar, UK.

…….. (2003). "Revoluciones Tecnológicas, Cambios de Paradigma y de Marco Socioinstitucional", en J. Aboites y G. Dutrénit (eds), *Innovación, Aprendizaje y Creación de Capacidades Tecnológicas*, UAM, México.

Phillips, E. y D. S. Pugh (2001). *Cómo Obtener un Doctorado. Manual para Estudiantes y Tutores.* Gedisa Edit., Barcelona.

Secretaría de Economía (2001-2006). Programa para el Desarrollo de la Industria del Software, PROSOFT. SE, México. www.economia.gob.mx

Steinmueller, E. W. (1996). "The U.S. Software Industry: An Analysis and Interpretative History", in D. Mowery (ed.), *The International Computer Software Industry*. Oxford Univ. Press, USA.

Vera-Cruz, A. O. (2004). *Cultura de la Empresa y Comportamiento Tecnológico.* Porrúa- UAM, Méx.

Von Hippel, E. (2000). "Horizontal innovation networks-by and for users". *Working Paper Num.* 4366-02, MIT Sloan School of Management, June, US.

…….. (2005). *Democratizing Innovation*. MIT Press, USA.

Yin, R. K. (1994). *Case Study Research: Design and Methods*. SAGE Publications, USA.